Prophesy: An Infrastructure for Analyzing and Modeling the Performance of Parallel and Distributed Applications*

Valerie E. Taylor, Xingfu Wu, Jonathan Geisler, Xin Li, Zhiling Lan
Department of Electrical and Computer Engineering
Celero Research Group, Northwestern University, Evanston IL 60208
Email: {taylor, wuxf}@ece.nwu.edu

Rick Stevens, Mark Hereld, Ivan R. Judson
MCS Division, Argonne National Laboratory, Argonne IL 60439

Abstract

Efficient execution of applications requires insight into how the system features impact the performance of the application. For distributed systems, the task of gaining this insight is complicated by the complexity of the system features. This insight generally results from significant experimental analysis and possibly the development of performance models. This paper presents the Prophesy project, an infrastructure that aids in gaining this needed insight based upon one's experience and that of others. The core component of Prophesy is a relational database that allows for the recording of performance data, system features and application details.

1. Prophesy Description

The Prophesy framework consists of three major components: data collection (left section), data analysis (right section), and three central databases, as illustrated in Figure 1. The data collection component focuses on the automated instrumentation and application code analysis at the level of basic blocks, loops, and functions. The entire code is instrumented at the basic block level such that a significant amount of performance information can be gathered to gain insight into the performance relationship between the application, the hardware and system software (e.g., compilers and run-time systems). The resultant performance data is stored in the performance database. The infrastructure also allows for performance data to come directly from users as well as the instrumented code.

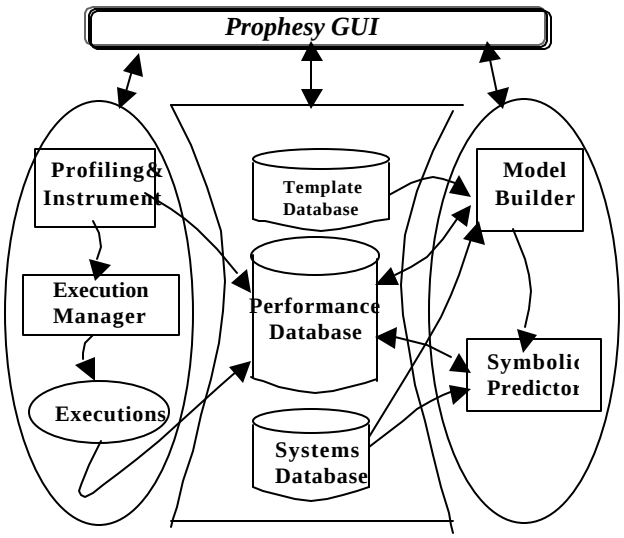

Figure 1. Prophesy framework

The data analysis component focuses on the analysis and modeling of an application. This component can be used to generate analytical performance models with coefficients, at the granularity specified by the user. The models are developed based upon performance data from the performance database, model templates from the template database, and system characteristics from the systems database. Optimization methods (such as linear or nonlinear least squares fit) are used to generate the coefficients for the analytical models. Prophesy allows for

* This research was supported in part by an NSF NGS grant EIA-9974960 and a grant from the NASA Ames HPCC Program.

the development of linear as well as nonlinear models. These models, when combined with data from the system database, can be used by the prediction engine to predict the performance of an application on a different compute platform.

The use of databases with Prophesy allows users to explore the performance models developed for different kernels, applications and systems. The data in the databases are organized in a hierarchical manner, allowing for the development of analytical models of different granularities. Prophesy is an infrastructure designed to explore the plausibility and credibility of various techniques in performance evaluation (such as scalability, efficiency, speedup, performance coupling between application kernels, etc.) and allow users to use various metrics collectively to bring performance analysis environments to the most advanced level.

2. Prophesy Database

The Prophesy Database is designed to facilitate the following query types:

- Identify the best implementation of a given function for a given system configuration (identified by the run-time system, operating system, processor organization, etc.)
- Use the raw performance data to generate analytical (nonlinear or linear) models of a given function or application; the analytical model can be used to extrapolate the performance under different system scenarios and can be used to assist programmers in optimizing the strategy or algorithms in their programs.
- Use the performance data to analyze application-system trends, such as scalability, I/O requirements, communication requirements, etc.
- Use the performance data to analyze user specific metrics such as coupling between functions.

The entities in the Prophesy database are organized into four areas: application information, executable information, run information and performance statistics. Descriptions of these four areas are given below.

- **Application Information**: includes one entity that gives the application name, version number and a short description. It is assumed that an application goes through various versions as one adds different functionalities over time. Data is placed into this entity when a new application is being developed.
- **Executable Information**: includes all of the entities related to generating an executable of an application. These entities include details about compilers, libraries and the control flow. It is assumed that applications may be developed using multiple languages, such as C, C++ and Fortran. Data is placed into these entities when a new executable is generated.
- **Run Information:** includes all of the entities related to running an executable, which includes the system information and inputs used for execution. This system may be a single processor, single parallel machine or distributed system. Data is placed into these entities for each run of a given executable.
- **Performance Statistics Information**: includes all of the entities related to the raw performance data collected during execution. Performance statistics are collected for different levels of granularities. Since a basic unit can be accessed from multiple functions, statistics are collected at the levels of both the function and basic unit.

3. Related Work

There exist some approaches to organizing performance data by using database techniques such as Snodgrass [1], SIEVE [2], and Performance Database Server [3]. However, most of these approaches only focus on performance data presentation. In contrast, Prophesy focuses on performance data representation, performance modeling, comparison and prediction.

4. Summary

This paper introduced the Prophesy project and its core component, the Prophesy Database. Prophesy is applicable to single processor, parallel machine and distributed system environments. The database is hierarchical, from which different granularities of performance information may be obtained. Currently, we are populating the database with performance information from some of the PACI applications. Our goal is to have the Prophesy database populated by results from numerous research groups, thereby allowing us to gain insights from each other's experiences.

References

[1] R. Hockney and M. Berry, *Public International Benchmarks for Parallel Computers*, PARKBENCH Committee: Report-1, February 7, 1994.

[2] S. R. Sarukkai, and D. Gannon, SIEVE: A performance debugging environment for parallel program, *Journal of Parallel and Distributed Computing* 18, 1993, 147-168.

[3] R. Snodgrass, A relational approach to monitoring complex systems, *ACM Transactions on Computer Systems*, Vol. 6, No. 2, May 1988, 157-196.